CS 576 – Assignment 1

1.a] 70.2 Mbps

Bit Rate = sampling \* quantization = (samples/second) \* (bits/sample)

Sub-sampling scheme is 4:2:0. So we have 4\*8 = 32bits for Y, 2\*8 = 16 bits for U, and 0 bits for V.

Thus, Avg bits per pixel = (32+16+0)/4 = 12

Bit rate = (450 \* 520 \* 12) \* 25 = 70,200,000 = 70.2 Mbps

1.b] 4.49 Gigabyte

Here, Sub-sampling scheme is 4:2:0 and we use 6 bits for Cr, Cb.

Thus, avg. bits/pixel = (4\*8 + 2\*6 + 0 \* 6)/4 = 44/4 = 11

Space required to store 10 min video = (450\*520\*11) \* 25 \* 600 = 38610000000 bits

= 38610000000 / (8\*1073741824) Gigabyte = 4.49 Gigabyte

2.a] Quantized sequence (if we consider signal values):

2.0, 2.25, 2.25, 3.25, 3.5, 3.5, 2.5, 3.0, 3.0, 3.0, 1.5, 1.0, 1.25, 1.25, 2.0, 2.25, 2.25, 2.25, 2.0, 2.5, 1.25, 0.25, -1.0, -1.0, -1.5, -1.0, -2.0, -1.5, -1.5, -0.5, 0.25, 1.0

Quantized sequence (if we consider the bucket number)

23, 24, 24, 28, 29, 29, 25, 27, 27, 27, 21, 19, 20, 20, 23, 24, 24, 24, 23, 25, 20, 16, 11, 11, 9, 11, 7, 9, 9, 13, 16, 19

2.b] We divide the interval [­4,4] into 32 uniformly distributed levels, so 5 bits are required to represent a level i.e. 5 bits to transmit a sequence and a total of 32x5 = 160 bits to transmit the sequence.

3.a] 7.5 rotations/sec.

Speed = 36 km/hr = 10 m/s

Diameter (D) = 0.4244 m

Distance covered in 1 rotation (circumference) = Pi \* D = 3.142 \* 0.4244 = 1.333 m

Rate of tire rotation in Rotations/second = 10/1.333 = 7.5 rotations/second

Thus the minimum frame rate needed to capture same rotation is 2x7.5 = 15 fps. (by Nyquist's theorem)

The recording was done at 24 fps > 15 fps. Thus we’ll see the same rotations per second as it was shot i.e. 7.5 rotations/sec

3.b] 0.5 rotations/second

Film rate is now 8 fps < 15 fps given by Nyquist’s theorem. So there will be aliasing.

The aliased frequency is given by = |R \* n - ­ fs|

R (film rate) = 8 fps

n = the closest integer multiple of the film rate and the rate of the signal being aliased = (8/7.5) = 1

fs = rate of signal being aliased = 7.5

Aliased frequency = | 8 x 1 -­ 7.5| = 0.5 rotations/sec.

Since there will be aliasing now, the wheel will look to rotate in backward direction.

Analysis Question 1:

I have used Peak-signal-to-noise ratio (PSNR) to analyze and quantify degradation as an error number. PSNR is usually expressed in logarithmic decibel scale. PSNR is defined by MSE (Mean Squared Error).

(Reference: <https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio>)

Typical values for the PSNR in image compression are between 30 and 50 dB, provided the bit depth is 8 bits, where higher is better. I plotted the values of Y, U, V from 2 to 120 by changing one value and keeping other two values constant at 1 (as given) and then created graphical visualization for the data as shown below:

Distortion Curve by varying Y from 2 to 120 and keeping U and V constant at 1:

Distortion Curve by varying U from 2 to 120 and keeping Y and V constant at 1:

Distortion Curve by varying V from 2 to 120 and keeping Y and U constant at 1:

The higher the value of PSNR indicates the higher quality of image. As seen in above plots, PSNR drops rapidly as we increase the value of Y, indicating that the image gets blurred at a faster rate as we increase Y than increase in U and V. This correlates with displayed image by program and I could observe the image gets blurred rapidly when Y increases above 3 or 4.

As we increase values of U and V the image quality also reduces but not as badly as in the case of Y. As U and V values are changed, there are patches introduced of red and yellow and become predominant at high values of U and V. We also get similar graph plots for other images.

Analysis Question 2:

As mentioned above, as we increase value of Y the image quality reduces drastically as compared to increase in values of U and V. So in the case of - (Y varying but U, V constant) we cannot improve the image quality. But in case of – (U varying but Y, V constant) or (V varying, but Y, U constant) I tried different techniques as below:

1. Interpolating missing value of U and V with mean.
2. Interpolating missing values of U and V with weighted average.

Below images outputs with mean and weighted average for Y=1, U=25, V=25, Q = 256

Image with interpolating U, V with Weighted Average:

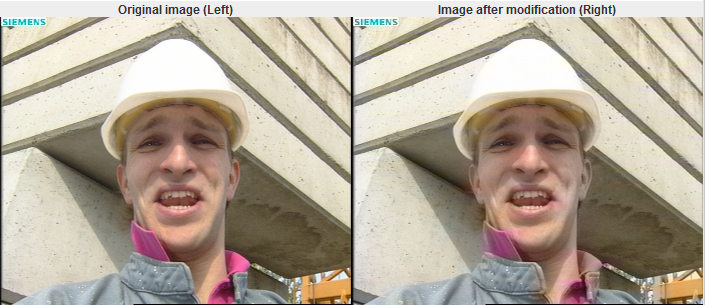
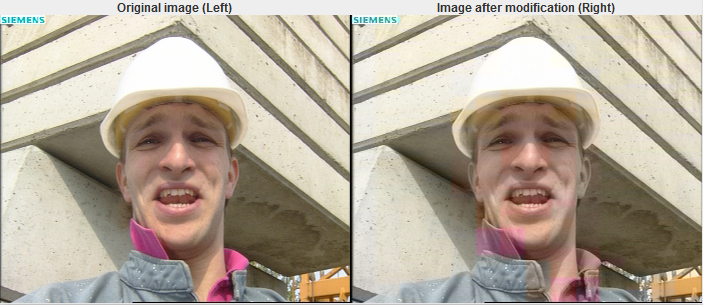


Image with interpolating U, V with Mean:



As we can see above, weighted average gives a better image quality as compared to mean, so I used weighted average it in my program. We can also use some other techniques like Seam Carving which remove the pixels with least energy to improve the image quality.